An integrated Expression Repository EHR system

Daniel Karlsson, Mikael Nyström, Bengt Kron
Project goals

- Develop and test a system for storing and querying pre- and post-coordinated SNOMED CT expressions in an Electronic Health Record (EHR) system
- SNOMED CT is used as a reference terminology
- SNOMED CT allows abstraction of EHR data for transfer to quality databases
Pre- and post-coordination
Pre-coordination: Pain

SNOMED CT Concept

Clinical finding

Neurological finding

Sensory nervous system finding

Pain / sensation finding

Pain

22253000 | Pain |
Pre-coordination: Pain

- 31 children
- 787 descendents
Post-coordination: Pain, Severity

SNOMED CT Concept

Qualifier value

Clinical finding

Ranked categories

Special atomic mapping values

Neurological finding

Grading values

Special disorder atoms

Sensory nervous system finding

Grades

Pain / sensation finding

Severities

Pain

Mild to moderate

22253000 | Pain | :
246112005 | Severity | =
371923003 | Mild to moderate |
Post-coordination: Pain, All 7 Severities
Post-coordination: Pain, all defining attributes

- Finding site, 26 000
- Associated morphology, 4 500
- Causative agent, 78 000
- Due to, 100 000
- After, 150 000
- Severity, 7
- Clinical course, 31
- Episodicity, 5
- Interprets, 26 000
- Has interpretation, 190

- Pathological process, 3
- Has definitional manifestation, 100 000
- Occurrence, 12
- Finding method, 53 000
- Finding informer, 4

- $5.5 \times 10^{44}$ possibilities
- Approximately $4.9 \times 10^{24}$ sensible possibilities
- 787 pre-coordinated descendents
Post-coordination: Pain after operation on neck

SNOMED CT Concept

Clinical finding

Procedure

Neurological finding

Procedure by method

Sensory nervous system finding

Surgical procedure

Pain / sensation finding

Operation on neck

Pain

After

Pain after operation on neck

Posttreatment pain

After

Postoperative pain

LIU EXPANDING REALITY
Post-coordination: Pain after operation on neck
Post-coordination:
Recurrent disease of cardiovascular system
Post-coordination:
Recurrent disease of cardiovascular system
Rationale for an expression repository

• Classifiers typically do not handle changes between releases and over time

• Each addition of a new post-coordinated expression creates a new version

• Unnecessary to classify each time an expression is submitted
  • Classifications of equivalent expressions can be reused

• Expressions might contain more characters than current systems can handle

• Fully defined post-coordinated expressions decrease the maintenance cost compared to extensions with pre-coordinated concepts
Expression repository architecture

- EHR prototype EEE
- XML DB
- Expression repository
- Data store
- OWL Classifier
Expression repository

• Classifier used to classify new post-coordinated expressions
  • OWL API-compliant classifiers (Snorocket, ELK, ...)
  • SNOMED CT expressions are translated to OWL
  • Result of classification fed to data store
• SNOMED CT query results aggregated from data store
Equivalent expressions

- Structural equivalent expressions
  - are only stored once in the repository
  - the equivalence can never be lost
- Semantic, but not structural, equivalent expressions
  - are stored individually
  - are linked to each other
  - have shared rows in the transitive closure table
  - the equivalence might be lost due to updates in new releases

85919009|Disorder of intestine|:
{ 363698007|Finding site|=66754008|Appendix structure|,
  116676008|Associated morphology|=23583003|Inflammation|}

85919009|Disorder of intestine|:
{ 116676008|Associated morphology|=23583003|Inflammation|,
  363698007|Finding site|=66754008|Appendix structure|}

128139000|Inflammatory disorder|:
{ 363698007|Finding site|=66754008|Appendix structure|}
Expression repository flow

1. Expression submitted
2. Parsing, syntactic normalizing
3. New expression?
   - Yes: Build OWL expression
   - No: Return id
4. Build OWL expression
5. Classify
6. Store
7. Return new id
Expression repository: Data store

- Data stored in tables in the relational database PostgreSQL
- Communicate through a Java class
- Contains a full release
- Manages the expression’s history
- Data storage and retrieval methods include
  - Store an expression
  - Store equivalence among expressions
  - Store parents and children to an expression
  - Get an expression’s parents
  - Get an expression’s ancestors
  - Get an expression’s children
  - Get an expression’s descendants
Expression repository: Data store
Transitive closure table

Current international full release gives 8,000,000 rows
Differences to Technical Implementation Guide

- Uses a full release
- Manages the expression’s history
- Uses a classifier
- Do not store the normal form